Name : W. K. D. D. Senuruk
Index Number : 190586H

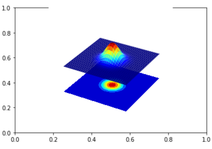
EN2550 Exercise 3 on Spatial Filtering

(1)

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import cv2 as cv

img = cv.imread(r'Images/butterfly.jpg', cv.IMREAD_REDUCED_GRAYSCALE_4)
cv.imshow('Image', img)
cv.waitKey(1000)

kernel = np.ones((9, 9), np.float32) / 81

img_box = cv.filter2D(src=img, ddepth=-1, kernel=kernel)
cv.imshow('Image', img_box)
cv.waitKey(1000)

fig, ax = plt.subplots(1, 3, figsize=(18, 6))
ax[0].imshow(cv.cvtColor(img, cv.COLOR_BGR2RGB))
ax[0].set_title('Original')

ax[1].imshow(cv.cvtColor(img_box, cv.COLOR_BGR2RGB))
ax[1].set_title('Box filtered')

gBlur = cv.GaussianBlur(img, (9, 9), 4)
cv.imshow('Image', gBlur)
cv.waitKey(1000)
cv.destroyAllWindows()

ax[2].imshow(cv.cvtColor(gBlur, cv.COLOR_BGR2RGB))
ax[2].set_title('Gaussian filtered')

for i in range(3):
    ax[i].set_xticks([]), ax[i].set_yticks([])

plt.show()
```

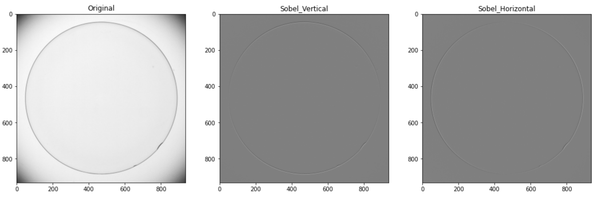
(2)

In [2]:
```
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm

fig, ax = plt.subplots()
ax = fig.add_subplot(111, projection='3d')

step = 0.1
x = np.arange(-5, 5 + step, step)
y = np.arange(-5, 5 + step, step)
xx, yy = np.meshgrid(x, y)
sigma = 1
g = np.exp(-(xx**2 + yy**2)/(2*sigma**2))

surf = ax.plot_surface(xx, yy, g, cmap=cm.jet)
cset = ax.contourf(xx, yy, g, zdir='z', offset=np.min(g) - 1.5, cmap=cm.jet)
ax.set_zlim(np.min(g) -2, np.max(g))

plt.axis('off')

plt.show()
```



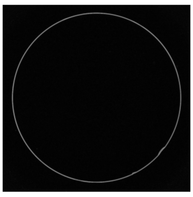
(3) (a)

In [3]:
```
img = cv.imread(r'Images/contact_lens.tif', cv.IMREAD_GRAYSCALE).astype(np.float32)
assert img is not None

sobel_v = np.array([[-1, -2, -1], [0, 0, 0], [1, 2, 1]], dtype=np.float32)
f_x = cv.filter2D(img, -1, sobel_v)
sobel_h = np.array([[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]], dtype=np.float32)
f_y = cv.filter2D(img, -1, sobel_h)

fig, ax = plt.subplots(1, 3, figsize=(18, 6))

ax[0].imshow(img, cmap='gray', vmin=0, vmax=255)
ax[0].set_title('Original')

ax[1].imshow(f_x, cmap='gray', vmin=-1020, vmax=1020)
ax[1].set_title('Sobel_Vertical')

ax[2].imshow(f_y, cmap='gray', vmin=-1020, vmax=1020)
ax[2].set_title('Sobel_Horizontal')

plt.show()
```



(3) (b)

In [4]:
```
grad_mag = np.sqrt(f_x**2 + f_y**2)
fig, ax = plt.subplots(1, 1, figsize=(18, 6))
plt.imshow(grad_mag, cmap='gray')
plt.axis('off')
plt.show()
```



(4)

In [5]:
```
image = cv.imread(r'Images/tom.jpg', cv.IMREAD_GRAYSCALE)
assert image is not None

kernel = np.array([[-1, -1, -1],
                   [-1, 9, -1],
                   [-1, -1, -1]])
image_sharp = cv.filter2D(src=image, ddepth=-1, kernel=kernel)
cv.imshow('Image', image_sharp)
cv.waitKey(1000)
cv.destroyAllWindows()

fig, ax = plt.subplots(1, 2, figsize=(12, 6))
ax[0].imshow(cv.cvtColor(image, cv.COLOR_BGR2RGB))
ax[0].set_title('Original_GrayScale')

ax[1].imshow(cv.cvtColor(image_sharp, cv.COLOR_BGR2RGB))
ax[1].set_title('Sharpened')

for i in range(2):
    ax[i].set_xticks([]), ax[i].set_yticks([])

plt.show()
```

In [6]:
```
image = cv.imread(r'Images/tom.jpg', cv.IMREAD_GRAYSCALE).astype(np.float32)

gaussian_1d = cv.getGaussianKernel(5, 2)
im_lp = cv.sepFilter2D(image, -1, gaussian_1d, gaussian_1d)
im_hp = image - im_lp
im_sharp = cv.addWeighted(image, 1.0, im_hp, 2.0, 0)

fig, axes = plt.subplots(1,4, figsize = (18,6))
axes[0].imshow(image, cmap = 'gray')
axes[0].set_title('original')
axes[0].axis('off')

axes[1].imshow(im_lp, cmap = 'gray')
axes[1].set_title('f_lp')
axes[1].axis('off')

axes[2].imshow(im_hp, cmap = 'gray')
axes[2].set_title('f_hp')
axes[2].axis('off')

axes[3].imshow(im_sharp, cmap = 'gray')
axes[3].set_title('Sharpened')
axes[3].axis('off')

plt.show()
```